Assignment 4: Fourier Approximations

Jnaneswara Rao Rompilli [EE20B052]

February 26, 2022

Abstract

- Fit two functions e^x and $cos(cos(x))$ using the Fourier series coefficients calculated by least squares fitting method.
- To plot graphs for better understanding

1 Introduction

The fourier function of a function $f(x)$ with period 2π can be calculated as follows

$$f(x) = a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + b_n \sin(nx) \tag{1}$$

where

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x) dx \tag{2}$$

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(nx) dx \tag{3}$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(nx) dx \tag{4}$$

2 Asssignment Tasks

2.1 Defining functions and plotting them

```
def expo(t):
    return exp(t)

def coscos(t):
    return cos(cos(t))

x_plot = generateSample(-2 * PI, 4 * PI)

plt.semilogy(x_plot, expo(x_plot), color="darkorange", label="Actual", linewidth=3)
plt.plot(x_plot, coscos(x_plot), color="darkorange", label="Actual", linewidth=3)
```

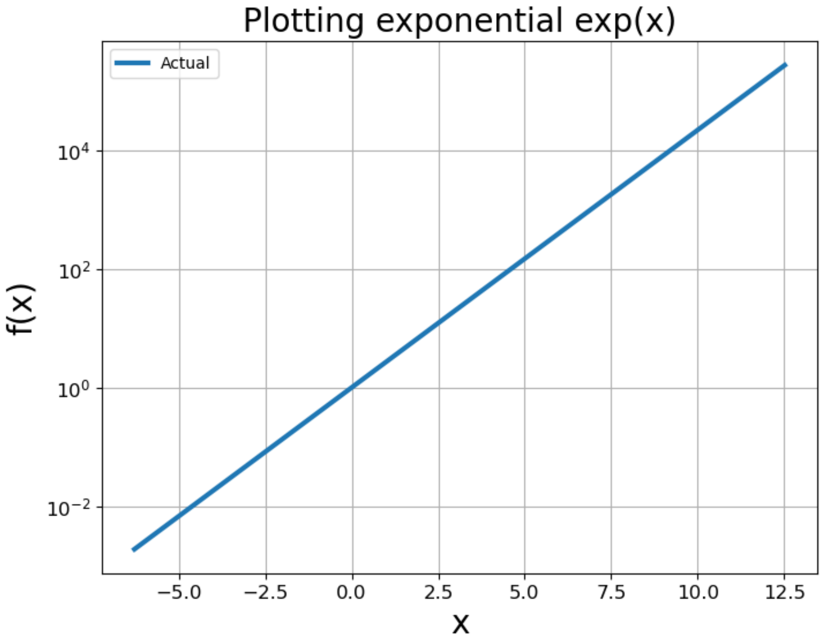

Figure 1: Plotting $exp(x)$ in semilogy scale

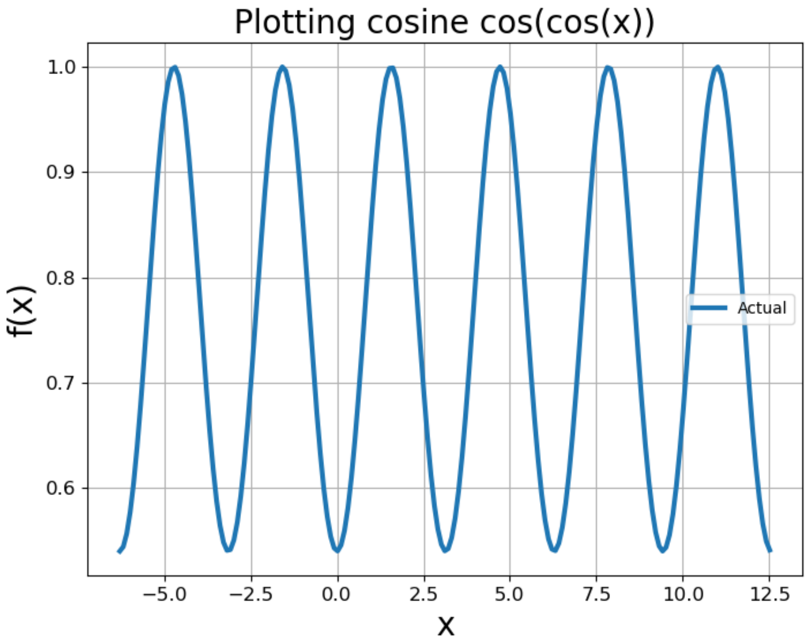

Figure 2: Plotting $cos(cos(x))$ in semilogy scale

The plots for $cos(cos(x))$ and $exp(x)$ are shown above

2.2 Computing fourier coefficients using integration

The first 51 coefficients are generated using the scipy.integrate.quad and the equations mentioned in the introduction function.They are saved in the following form $\begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ \vdots \\ a_{25} \\ b_{25} \end{bmatrix}$

```
an_exp = integrate.quad(a_expo, 0, 2 * PI, args=(0))[0] / (2 * PI)
an_coscos = integrate.quad(a_coscos, 0, 2 * PI, args=(0))[0] / (2 * PI)

coeff_exp[index][0] = an_exp
coeff_coscos[index][0] = an_coscos
index += 1

for k in range(1, 26):
    an_exp = integrate.quad(a_expo, 0, 2 * pi, args=(k))[0] / pi
    bn_exp = integrate.quad(b_expo, 0, 2 * pi, args=(k))[0] / pi
    an_coscos = integrate.quad(a_coscos, 0, 2 * pi, args=(k))[0] / pi
    bn_coscos = integrate.quad(b_coscos, 0, 2 * pi, args=(k))[0] / pi

    coeff_exp[index][0] = an_exp
    coeff_exp[index + 1][0] = bn_exp
    coeff_coscos[index][0] = an_coscos
    coeff_coscos[index + 1][0] = bn_coscos

index += 2
```

- Fourier coefficients of an even function should be zero. As expected b_n is close to zero for $cos(cos(x))$ as it is an even function.
- The coefficients in a fourier series represent what are the frequencies happen to be in the output. The function cos(cos(x)) doesn't contain many different frequencies, so the values decay out quickly whereas the the exponential function is combination of different frequencies
- The loglog plot is linear for e^t because Fourier coefficients of e^t decay with $1/n$ or $1/n^2$. The semilog plot is linear in $cos(cos(t))$ as the Fourier coefficients decay exponentially with n

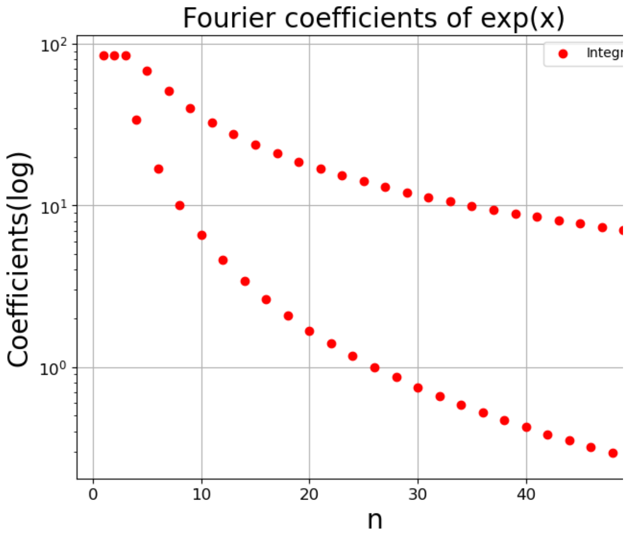

Figure 3: semilogy scale

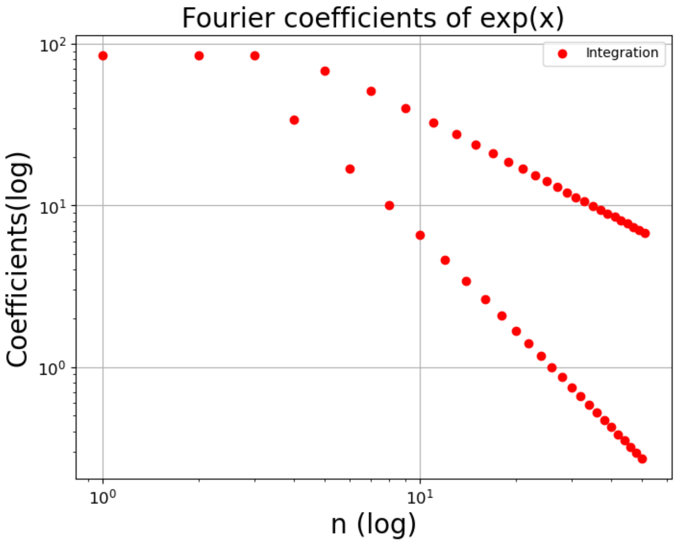

Figure 4: loglogy scale

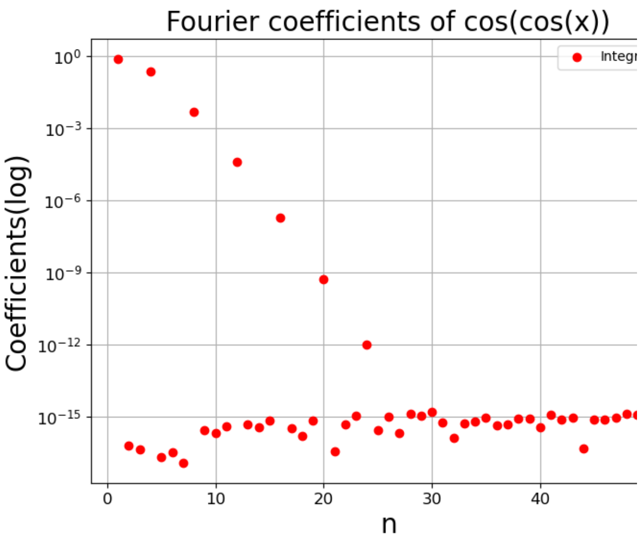

Figure 5: semilogy scale

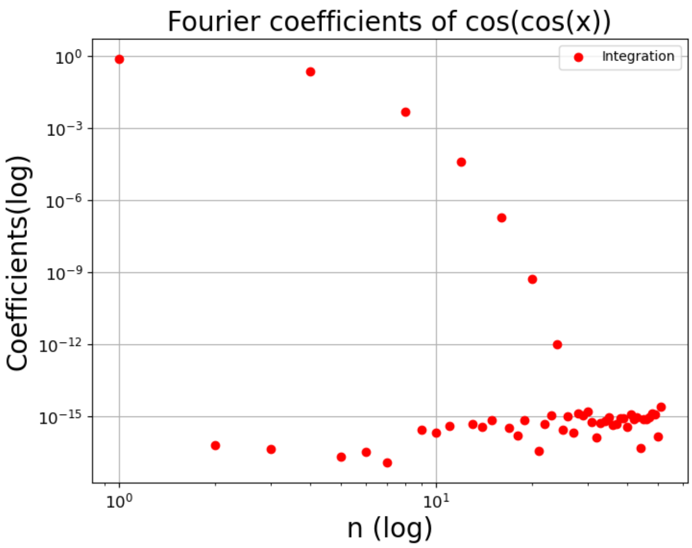

Figure 6: loglogy scale

2.3 Using Least squares approach

Now, we used Least Squares approach to find the Fourier series coefficients. We linearly choose 400 x values in the range $[0, 2\pi)$ and calculated coefficients using scipy.lstsq function

```
count = len(x)
M = np.zeros((count, 51))
M[:, 0] = 1
for k in range(1, 26):
    index = 2 * k
    M[:, index - 1] = cos(k * x)
    M[:, index] = sin(k * x)

coefflst_coscos = lstsq(M, y_coscos, rcond=None)[0].reshape((-1, 1))
coefflst_exp = lstsq(M, y_exp, rcond=None)[0].reshape((-1, 1))
```

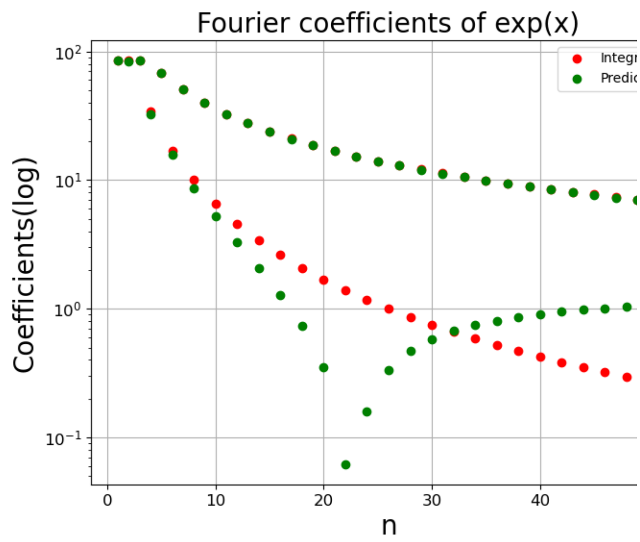

Figure 7: semilogy scale

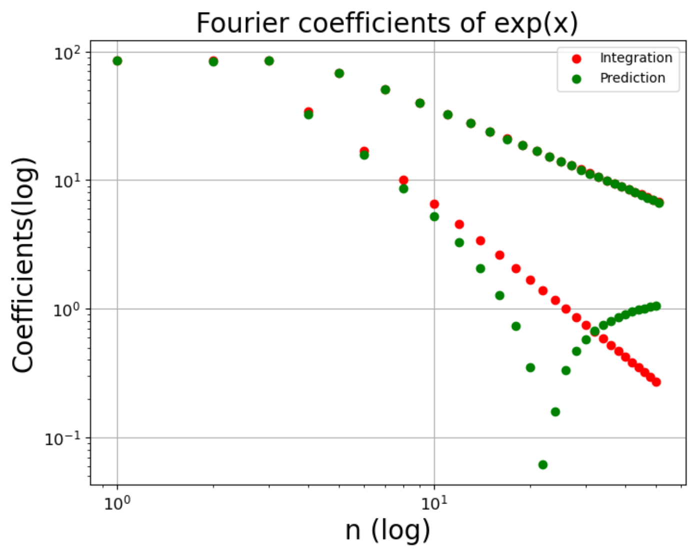

Figure 8: loglogy scale

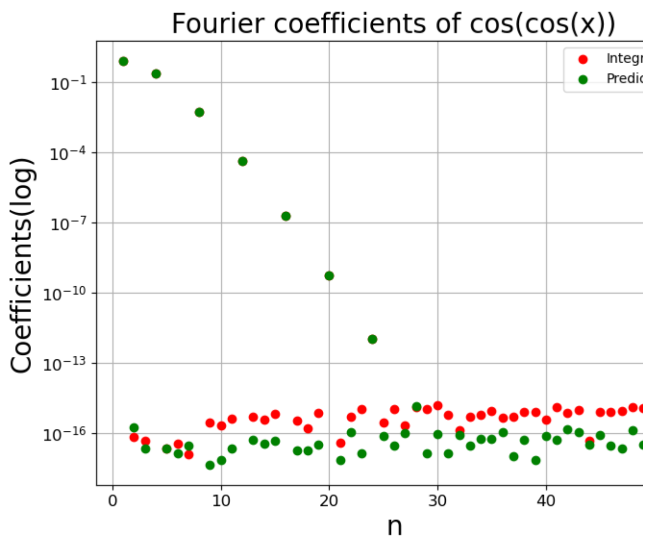

Figure 9: semilogy scale

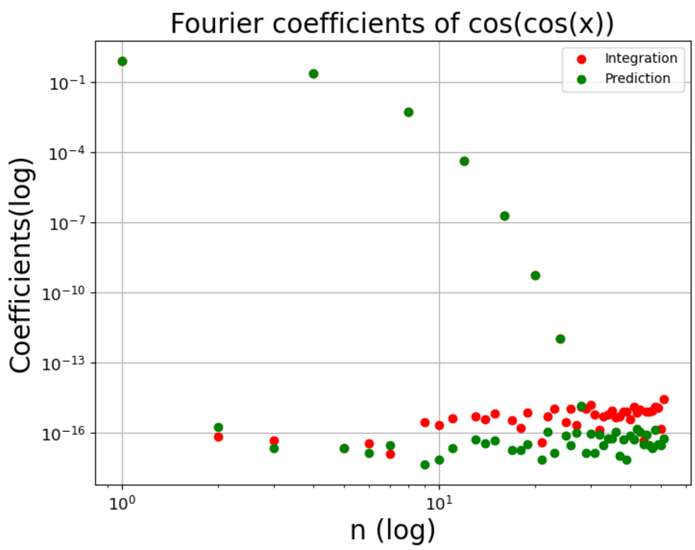

Figure 10: loglogy scale

The numpy linalg lstsq() solves the equation ax = b by computing a vector x that minimizes the Euclidean 2-norm $|b–ax|^2$

- Maximum deviation for $exp(x)$ is 1.3327
- Maximum deviation for $cos(cos(x))$ is 2.6607e-15

We can observe error in $exp(x) >> cos(cos(x))$

2.4 Plotting result

The deviation in more in $exp(x)$ fitting beacuse fourier series exists only for periodic functions but e^x is a non periodic function

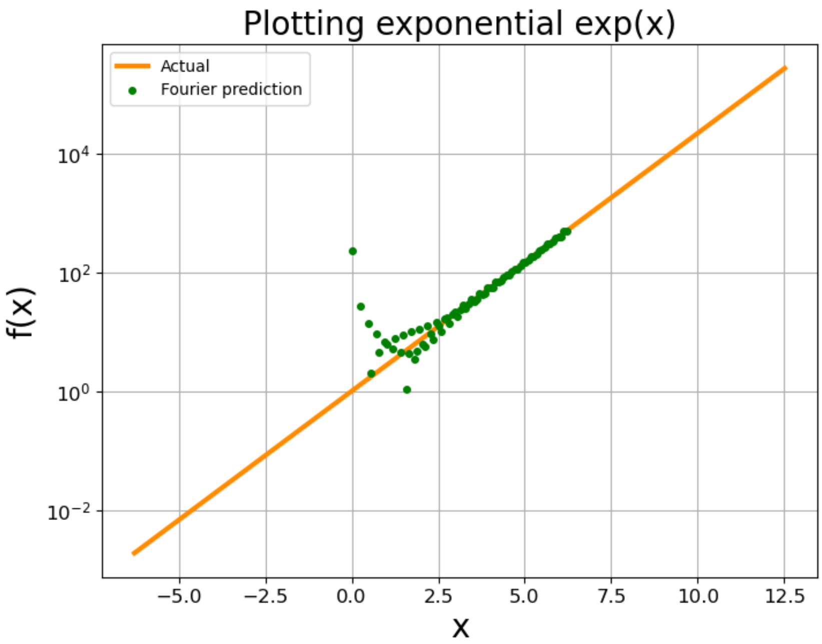

Figure 11: Actual and predicted values of $exp(x)$

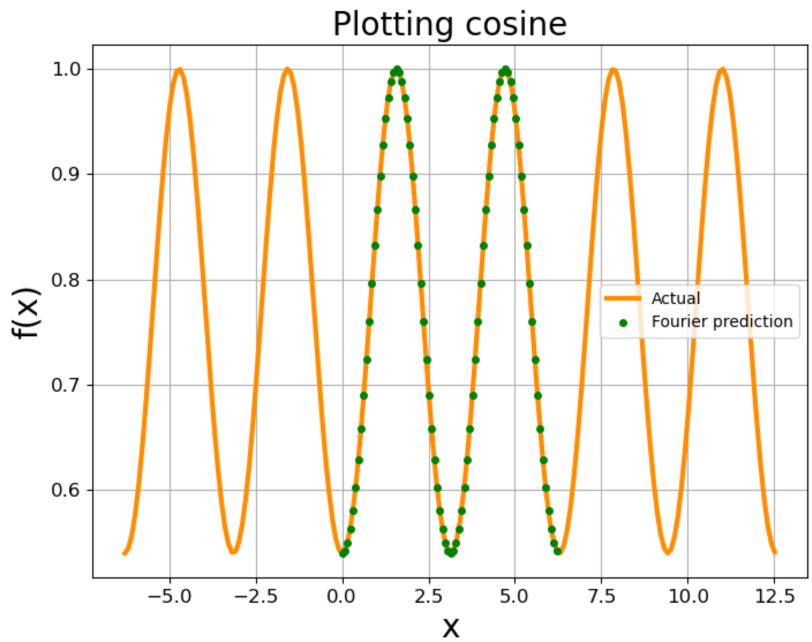

Figure 12: Actual and predicted values of $cos(cos(x))$

3 Conclusion

We computed Fourier series coefficients using two different methods

- Using integration
- Least Square Fitting method

We found close matching in two methods in case of $cos(cos(x))$ while, there is a larger discrepancy in $exp(x)$